

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Request for Rehearing Under 37 CFR §
41.52

Ramos et al.

Art Unit 2174
Conf. No.: 6353

Application No.: 09/636,102

Filed: August 10, 2000

Examiner: T. Vu

For: WATERMARK ENCODER AND
DECODER ENABLED SOFTWARE
AND DEVICES**Via Electronic Filing**

Date: July 31, 2006

REQUEST FOR REHEARING UNDER 37 CFR SECTION 41.52

This Request for Rehearing is responsive to the Decision of the Board mailed May 31, 2006, affirming the rejection of claims 2, 5-7, 9 and 14-20 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 5,606,609 to Houser et al. (Houser), and the rejection of claims 11-13 under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 5,801,689 to Huntsman (“Huntsman”) in view of Houser. Please charge any fee required for this rehearing to deposit account 50-1071.

Requests for Rehearing under 37 CFR Section 41.52 are permitted where they “state with particularity the points believed to have been misapprehended or overlooked in rendering the decision.” Appellants respectfully submit that the decision: (I.) misapprehended the interpretation of the term “watermark” and (II.) overlooked certain claim elements that distinguish the claims over the Board’s interpretation of the cited art.

Regarding point I, for claims 2, 9, 10, 12, 15 and 20, the Board’s decision misinterpreted the meaning of the term “watermark” in these claims and applied distinctly different aspects of Houser, namely the “security object” and the “watermark” in the cited Houser reference to the term “watermark” in the claims, which is inconsistent and incorrect, whichever one is applied.

Regarding point II, Appellants set forth a claim by claim analysis of claim elements that they indicated were absent from the cited art, yet were not addressed in the Board's decision.

I. The claimed “Watermark” vs. alleged teachings of the cited art

Because point I. impacts several of the claims, we address it first, and we then address point II.

Appellants believe that the Board's interpretation of “watermark” includes inconsistencies that, at least, prevent clear understanding of the decision. Thus, this needs to be addressed prior to addressing the other points.

The Board's Decision is incorrect in stating that: “the references teach the use of the traditional digital watermark capability consistent with appellants' understanding and definition of this term at pages 1 and 2 of the specification...” because:

1. The Decision explains on page 3 that Houser's “security object” corresponds to the claimed watermark, and then on page 4 relies on Houser's reference to “watermark generators” in Figs. 8 and 10 as teaching the claimed watermark, which is inconsistent and contradicts the conclusion that Houser teaches “the traditional digital watermark capability.”
2. Both Houser's security object and its “watermark” fail to correspond to the claimed watermark as properly interpreted.

Houser's Security Object vs. Houser's Watermark

Houser's security object is an object “embedded” in an electronic document file in the context of object linking and embedding technology (OLE), which is different than the application's use of the term “watermark.” In the context of OLE, an “embedded” object relates to a mechanism within an electronic document file that invokes an executable function (computer program), which provides additional processing capability to the “host” application program used to create and edit the document. This allows the host application program to display other types of data in the document without having to manually invoke the separate program. In Houser's implementation, this executable function is a signature interpreter module invoked to verify aspects of an electronic document and display/print a signature graphic. As the Board noted, the security object includes a security object identifier that identifies the file containing a security object interpreter for processing security information of the security object.

This identifier serves to identify where the security object program resides so it can be launched automatically by the OLE controller.

In contrast, Houser's "watermark" is a visible graphic that is displayed or printed. In Houser, this graphic may be generated using the extracted security information and/or may be a time-dependent graphic. See, for example, col. 4, lines 52-60; col. 5, lines 56-65. Houser's watermark is not embedded in the document and does not include an identifier.

Thus, it is incorrect to contend, on the one hand, that Houser teaches some traditional digital watermark capability when, on the other hand, the Board interprets distinctly different elements of Houser to establish a so-called "traditional digital watermark capability." Appellants respectfully submit that the Board must articulate a consistent understanding of "watermarking capability" before it can be deemed to be "traditional." In fact the differences among the security object in Houser, the watermark in Houser, and the properly interpreted watermark in the claims at issue clearly show that there was no traditional notion of digital watermark capability in the cited art.

Houser's Security Object vs. Claimed Watermark

At least a portion of the Board's decision relies on the allegation that Houser's security object corresponds to the claimed watermark. The specification defines a watermark to include at least the following attributes: it is a machine readable code embedded in media; see, e.g., page 1, lines 22-35; and it is embedded in a media signal by altering the host media signal (such as an audio, image or video signal); see, e.g., page 2, lines 2-5. Under the application's definition, the media signal itself is modified to carry the code, rather than storing additional data like Houser's security object separate from the media signal in the file.

Appellants agree that Houser's security object includes machine readable data. Yet this data is not embedded in a media signal by altering the media signal. In Houser, the data that forms the security object is placed in a document file at discrete locations, but it does not alter any media signal that resides in that file. In fact, Houser teaches that the presence or absence of security objects in the file should not influence the hash calculation of other data in the file, and these objects are placed in the file so as not to disturb other data in the file. See col. 12, lines 30-40. While the Board correctly notes that Houser's document file may include media signals like images or sound, Houser's security object is not embedded in these signals by altering the text,

video, images or sound in the file. In sum, Houser's security object is not a "watermark" in the same way this term is defined and claimed in the patent application at issue because it is not embedded by altering a host media signal or signals in the file. Therefore, Appellants respectfully request the Board to reconsider its conclusion near the bottom of page 3 of the decision that the security object alters host media signals.

Houser's Watermark vs. Claimed Watermark

Houser's watermark is a graphic that is not embedded in the document file. Therefore, it too, does not correspond to the claimed watermark.

In addition, the Board asserts on page 4 that since Figs. 8 and 10 show "watermark generators," these watermark generators "of necessity require the ability to decode the respective watermarks to be able to derive the underlying embedded information therein." Respectfully, this is a fundamental misunderstanding of the watermark in Houser. The watermark in Houser is a graphic that is printed or displayed. See, for example, col. 4, lines 52-60; col. 5, lines 56-65. That is all it is. There is no need to decode anything from it because it is simply intended to be a graphic or video (e.g., a moving graphic) that a user visually observes in the displayed or printed form to get some indication that the file has passed verification by the signature interpreter module. See col. 17, lines 5-13. Since the Board's decision rests, at least in part, on this misapprehension, the parts of the decision that rely on it must be withdrawn.

II. The Board's Decision Overlooked Appellants Arguments of Claim Elements

Absent from the Cited Art

Since Appellants argued the claims independently, the following addresses each in a claim by claim analysis. Even if one accepts the Board's handling of the term "watermark," Appellants believe that the Board has failed to address important arguments establishing that the cited art does not teach all of the elements of the claims.

Claim 2

In addition to Houser not teaching the claimed watermark, the Board has not refuted Appellants argument that Houser fails to teach "a file browser extension for decoding an object identifier from a selected media object file and displaying in an extension of the user interface metadata or an action associated with the media object file via the object identifier."

On page 6 of the Decision, the Board states that: “As with the subject matter of claim 2 and all other claims on appeal relating to this feature, the claimed file browser is not argued by appellants in the brief and reply brief not to be present in Houser.”

In fact, Appellants have clearly argued that the cited art does not teach certain aspects of the user interface displayed by the file browser in claim 2. The claimed file browser recites: “a file browser for displaying **in a user interface** a representation of media objects stored in memory.” Appellants emphasized in the reply that Houser does not teach the file browser extension “for decoding...and for displaying **in an extension of the user interface** metadata or actions...[emphasis added],” which expressly refers to an extension of the user interface displayed by the file browser. Rather than argue every element, Appellants focused the Board on the extension of the user interface, and in fact, did emphasize this aspect of the user interface displayed by the file browser in the novel combination of elements in claim 2. However, the Board’s decision does not address this argument and fails to establish that Houser discloses this aspect of claim 2.

On page 5 of the Decision, the Board refers to security information, including among other things a signature chop, which it asserts relates to the recited metadata, but this information is not displayed by a file browser extension in an extension of the user interface displayed by the file browser. The signature interpreter module displays the signature chop in the user interface of the host application, not the user interface displayed by the file browser. Neither the application program, nor the signature interpreter module provides an extension of the user interface displayed by a file browser as claimed. Therefore, Houser does not anticipate claim 2.

In addition, the security object identifier, which the Board contends corresponds to the claimed object identifier, is used for launching verification processing, and specifically, for identifying the file that contains the signature interpreter module capable of processing the security information in the security object. Houser’s identifier is not used to associate Houser’s alleged metadata (security information and signature chop) with Houser’s alleged media object (the electronic document file). As the Board cited in cols. 21 and 22, the process for invoking document verification and electronic chop display/printing is as follows: the application program that creates the electronic document encounters an OLE security object, it uses the identifier in this object to invoke the signature interpreter module, which in turn, performs

verification of the security information just encountered and displays/prints an electronic chop as appropriate. Further, the data that gets displayed is not associated with the document via the security object identifier because the application program already has encountered the security object, including its security information, and uses the security object identifier to launch the signature interpreter module to process the security information that the host application already has encountered. See col. 22, lines 10-27.

Appellants asserted that these elements of displaying metadata or an action associated with the media object file via the object identifier in an extension of the user interface displayed by the file browser were absent from Houser, yet the Board did not address them in its decision.

Claim 5

In the Reply brief, Appellants took issue with the Office's position that the verification processor corresponds to the claimed metadata server. Appellants then further argued: "Even assuming that the verification processor corresponds to the claimed metadata server, Houser fails to teach forwarding an object identifier to the verification processor, and displaying metadata or an action returned from this processor in an extension of the user interface of the file browser as claimed."

The Board's decision does not clarify which teachings, if any, of Houser are alleged to correspond to the metadata server. In addition, the Board's decision incorrectly states that Appellants did not argue aspects of the file browser. Appellants clearly argued that Houser does not teach the claimed extension of the user interface displayed by the file browser in combination with other elements.

As noted above, Houser's identifier is used to identify to the OLE controller the file which contains the "signature interpreter module" (the verification processor 830 is a part of the signature interpreter module as shown in Fig. 8, col. 15, lines 54-60). The OLE controller maintains an association between the identifier and the signature interpreter module. See col. 21, line 60 to col. 22, line 27 (which is the same passage cited by the Board). Claim 5 recites that the file browser extension forwards the identifier to the metadata server, which returns metadata or an action displayed by the file browser extension in the user interface displayed by the file browser. In Houser, the host application forwards the security object identifier to the OLE controller, which identifies the signature interpreter module file and then passes control to it.

The host application is not a file browser extension as claimed because it does not display metadata or an action in an extension of the user interface displayed by a file browser. The OLE controller does not return metadata or an action for display, but instead only invokes the signature interpreter module identified by the security object identifier. Finally, the verification processor and the signature interpreter module (which includes the verification processor 830 as shown in Fig. 8) do not correspond to the metadata server because they are not forwarded the identifier from a file browser extension as claimed.

Appellants previously argued that Houser does not teach the metadata server and the extension of the user interface of the file browser in combination with other elements. The Board has not, and based on the teachings of Houser, cannot refute this argument, and thus, must reverse the rejection.

Claim 6

The Board has not addressed claim 6 at all, even though the elements of this claim were independently argued. Thus, the rejection of this claim must be reversed.

Claim 7

As with claim 5, the Board appears to have overlooked Appellants argument that Houser fails to teach the “file browser extension... for displaying in an extension of the user interface” displayed by the file browser. The user interface refers to the user interface displayed by the file browser, and Houser fails to teach this aspect of claim 7 for similar reasons as provided above.

The Board’s decision attempts to address Appellants’ argument that Houser does not teach: “the metadata or action is displayed as a URL link to information or a program associated with the selected media object file” in combination with the other elements of claim 7. The rejection at issue is whether Houser anticipates claim 7. The Board’s decision makes reference to use of the term Internet in Houser, yet none of this disclosure of Houser discloses these elements of claim 7. The Board contends that “the invocation or use of the internet necessarily requires the use of an internet browser to the extent recited in independent claim 7 on appeal.” As a general matter, there are many forms of data transfer that use the internet without the use of an internet browser, and Houser’s certainly does not disclose the cited claim language expressly or inherently. But more specifically, claim 7 refers to a particular aspect of the display in the extension of the user interface, namely display of the metadata or action as a URL link to

information or a program. This particular aspect is clearly not disclosed in Houser, and therefore, the 102 rejection is unsupportable.

The Board then contends that Appellants did not argue motivation to combine Houser and Huntsman for claims 7 and 11, yet the combination of Houser and Huntsman is only applied to claims 11-13. If the Board is attempting to assert a new ground of rejection, it should expressly state it so that Appellants can respond appropriately.

Claim 9

The Board has not addressed Appellants' argument that Houser fails to teach: "a file browser extension"..."displaying in an extension of the user interface one or more options for enabling a user to enter input to control the encoding of the object identifier" as set forth in claim 9. Appellants have already noted the significance of the extension of the file browser user interface and Houser's lack of teaching in this regard. Further, Houser does not teach these aspects of the claimed user interface relating to options for encoding the identifier.

Again, to reiterate the above argument regarding Houser's failure to teach the claimed "watermark," Houser does not teach the claimed watermark encoder. The Board appears to refer to the "watermark generator" and the signature insertion module in Houser. These are distinctly different and unrelated elements of Houser, and neither corresponds to the claimed watermark encoder. The signature insertion module may include a security object identifier, but it fails to display one or more options for enabling the user to enter input to control the encoding of the object identifier with a watermark encoder as claimed. Houser's watermark generator generates a graphic for display/printing as part of verification, not insertion, and does not correspond to these elements of claim 9 either.

Claim 10

Houser fails to teach the "extension...for decoding a watermark... and displaying...associated with the media object file via the watermark" in the novel combination of claim 10. Appellants submit that the Board has incorrectly interpreted "watermark." The two aspects of Houser cited as the claimed "watermark," namely the security object and Houser's watermark do not correspond to the watermark in claim 10 as explained previously.

Claim 11

The Board appears to have missed the fact that claim 11 recites: “**for inserting a handler** into the document **when an object identifier is extracted** from the media object...” in combination with the other elements. Here, the Board notes the passage at col. 7, line 29, which teaches that the security object identifier may be executable code that invokes verification processing. Houser is completely silent about “**inserting a handler in the document when an object identifier is extracted from the media object.**” The cited passage in Houser merely means that rather than use a string or number to identify the signature interpreter module and then launch it, the identifier can comprise executable code to launch verification processing.

Claim 11 recites a listener program for inserting a handler when the object identifier is extracted. There is no analogous teaching in Houser. In particular, whether Houser’s identifier is a number or executable code, there is no mechanism analogous to the claimed listener program that inserts a handler into the document when the security object identifier is “extracted” as claimed. Houser clearly does not insert anything analogous to the claimed handler, which is “operable to display metadata linked via the object identifier in response to user input,” when the identifier is read from the document file. Houser proceeds to launch verification and display of a chop when the host application encounters a security object identifier in a document file, but Houser does not insert a handler into the document with the claimed attributes of the handler.

Appellants did assert that there is no motivation to combine Houser and Huntsman at page 9 of the Appeal Brief. Even when combined, these references fail to teach all of the elements of the claims. Therefore, one of skill in the art could not combine them in a manner that would yield the browser of claim 11. There is no need to elaborate on the lack of motivation to combine when the combination lacks all of the claim elements. Appellants did not forfeit this motivation to combine argument.

Claim 12

The claimed “watermark” further differentiates this claim from the cited art as described previously.

Claim 13

The Board contends that Houser’s security object identifier corresponds to the claimed object identifier, and further notes that Houser’s identifier may be executable code that invokes verification processing. Regardless whether Houser’s identifier is implemented as a string,

number or executable code used to launch verification processing, it is not used to retrieve metadata from a metadata server as claimed by sending the object identifier to a metadata server. If one applies Houser's teaching that the security object identifier is a form of executable code to launch verification processing of the security object, then Houser is even further away from the claimed invention because this executable code is not sent to a metadata server to retrieve metadata. It is merely executed to launch verification processing of the security information in the document file.

If, on the other hand, one applies Houser's teaching that the security object identifier is a number or string used by the OLE controller to find the file of the signature interpreter module, then Houser is still not relevant because this identifier is not sent to a metadata server to retrieve metadata from the metadata server as claimed. Claim 13 recites that the metadata retrieved from the metadata server is displayed by the handler. Yet, in Houser, the only thing that is even arguably retrieved using the identifier is the signature interpreter module. The other teachings of Houser cited by the Board as metadata, such as the security information and the signature chop, are not retrieved from a metadata server by sending the identifier to the server because these items are already part of the security object in which the security object identifier resides. See col. 22, lines 10-27. Once Houser has the security object identifier, it already has the security information, and it does not retrieve metadata as claimed from a metadata server.

Huntsman does not redress the deficiency of Houser and is not relied on in this regard.

Claim 14

Even if one accepts the Board's broad interpretation of "brand identifier," none of the information in Houser that is broadly alleged to constitute a brand identifier is obtained by the recited method in claim 14. The Board alleges that Houser's security object identifier corresponds to the claimed object identifier, but this identifier is not sent to a metadata server, and the alleged "brand identifier" in Houser is not received from the metadata server. An extensive electronic search through Houser reveals that every reference to identifier in Houser refers to using the identifier to invoke verification processing. This identifier is not sent to a metadata server as defined in claim 14. In addition, the graphic elements alleged to constitute the brand identifier in Houser are not received from a metadata server to which the identifier is sent as claimed. Therefore, Houser does not anticipate claim 14.

Claim 15

Houser's alleged identifier is not decoded from a watermark embedded in a media object as set forth above.

Claim 16

The Board contends that Houser's printed chop corresponds to the claimed superimposed graphic. Yet as noted, this printed chop is not received from a metadata server as claimed. In addition, claim 16 recites that the media object is a video or image, and in reference to base claim 14, the object identifier is decoded from the video or image. In contrast, Houser's alleged identifier is not decoded from video or an image, but instead, is separate from any video or image information in the document file.

Claim 17

The electronic chop in Houser is not a hot link to information or an action as claimed.

Claim 18

Selection of Houser's electronic chop does not cause retrieval of the information or action from a remote server as recited in claim 18.

Claim 19

The Board's position appears to be that Houser's security object identifier corresponds to the claimed object identifier, yet this identifier is not used to look up metadata, where the representation of the metadata is included in the user interface of the media player as claimed.

Claim 20

Houser fails to teach decoding the object identifier from a watermark embedded in the media object as described previously.

For the above reasons, Appellants respectfully request the Board to reconsider its position and reverse the rejection of the claims. If the Board intended to issue a new ground of rejection, it should so state so that Appellants' options for responding to such a rejection are clearly available to it. Nevertheless, the combined teachings of Houser and Huntsman fail to teach all of the elements of the claims, and as such, the most appropriate action is to reverse the rejection.

Date: July 31, 2006

CUSTOMER NUMBER 23735

Phone: 503-469-4800
FAX 503-469-4777

Respectfully submitted,

DIGIMARC CORPORATION

By



Joel R. Meyer
Registration No. 37,677